

Appendix D

Extensions to the Object-Relational Mapping System of the Present Invention

A. Extension for Security (login and access rights)

1: Specifying login rights for users with respect to an ORM Specification

Currently any user who can login to the database can access database elements and also the objects provided through the current invention of object-relational mapping. However, this access may be further restricted by providing a login mechanism at the level of ORMId identifying a particular ORM Specification. This would enhance the security at the higher level whereby the number of users may be restricted at the level of object domain of an ORM Specification identified by an ORMId.

Implementation details:

This feature may be implemented by keeping a table with the following 3 columns:

ORMId	varchar(80)
userName	varchar(80)
password	varchar(80)

The Object Administrator owns this table and maintains entries in this table as per security requirements.

1
2 Essentially, at the time of opening a service with an ORMId specification, the supplied
3 user name and password would be checked against this table to determine if the user is
4 authorized to use the ORM Specification.

6 **2: Specifying access rights for users with respect to an ORM Specification**

7 Currently access rights (read, write etc.) on databases elements (tables, views,
8 stored procedures) may be defined for users accessing the database. However, if
9 relational data is exposed as objects then the current mechanism does not lend itself
10 well in terms of defining access rights based on object definitions. With ORM
11 specification mechanism of the current invention, it would be very convenient and
12 innovative to add access rights at 2 levels:

13 1: High level - Access rights are granted at the level of ORMId. A user may be assigned
14 read or write access for a particular ORM specification identified by an ORMId and
15 here are some possible rules:

- 16 • If the user has not been given any access rights, the user cannot use Database
17 Exchange Unit for that ORM Specification.
- 18 • If the user has been given only read access, the API calls for insert, update, delete
19 and SQL Statement cannot be executed by that user.
- 20 • Otherwise, the user can execute all API calls for all the classes identified in the
21 ORM Specification.

2: Low level - Access rights are granted at the level of ORMId and a class within the ORM specification identified by the ORMId. The rules as above apply however the restrictions are observed at individual class levels.

Grammar for specifying access rights for objects:

Some of the Define new rules in ORM Grammar to specify the name of a stored procedure and the class of the objects which can be constructed using the returned values from the execution of the stored procedure. Here is an example of a rule:

```
<ACCESS-RIGHT> ::= ACCESS_RIGHT <USER-NAME> ACCESS_TYPE {READ |  
WRITE | EXECUTE} [<CLASS-NAME>]
```

For example:

```
ACCESS_RIGHT dPeriwal ACCESS-TYPE WRITE
```

or

```
ACCESS_RIGHT jSmith ACCESS-TYPE READ Employee
```

The first specification grants all accesses to the user dPeriwal. The second one grants just the READ access on only the Employee class objects to the user jSmith.

Implementation details:

- ORM Data Structure Creation Unit will create a structure in memory for each such

1 specification and during the execution of the API call, this data structure would be
2 consulted to execute the stored procedure, create objects from the stored procedure
3 result set and return the objects to the application.
4
5

6 **B. Extension for Relational Stored Procedures**

7 **Mapping for stored procedures**

8 Currently stored-procedures are used to define programs which can be executed
9 in the relational database engine itself providing better performance for executing some
10 business logic. Often times the output of these stored procedures are returned to the
11 application program in the form of a single value or a set of values (as multiple rows).
12 It would be advantageous for an object-oriented program to invoke stored procedures
13 for performance reasons and get the returned values as program objects. So the current
14 invention can be enhanced in the following manner:
15

16 **Grammar for specifying mapping for stored procedures:**

17 Define new rules in ORM Grammar to specify the name of a stored procedure and the
18 class of the objects which can be constructed using the returned values from the
19 execution of the stored procedure. Here is an example of a rule:

20 `<SP-SPEC> ::= SP_CLASS <CLASS-NAME> STORED_PROC`
21 `<STORED-PROCEDURE-NAME>`

1 For example:

2 SP_CLASS abc STORED_PROC proc1

3 The above specification states that the output of the stored procedure "proc1" should be
4 used to produce objects of class "abc"

5

6 **API for invoking stored procedures:**

7 An object-oriented program can use the following API to invoke a stored procedure and
8 retrieve resulting objects. The API is similar to query API:

9 Vector storedProc (String spName, Vector parameters, long maxObjects, long spFlags)
10 where parameters are used to pass input values to the stored procedures.

11

12 **Implementation details:**

- 13 • ORM Data Structure Creation Unit will create a structure in memory for each such
14 specification and during the execution of the API call, this data structure would be
15 consulted to execute the stored procedure, create objects from the stored procedure
16 result set and return the objects to the application.

17